

# # Face Recognition Attendance System - Project Report

## 1. Introduction

In modern educational institutions and workplaces, manual attendance marking is time-consuming and prone to errors. Traditional methods like paper-based attendance or manual entry into digital systems can lead to inefficiencies. To address this, we propose a **Face Recognition Attendance System**, which uses facial recognition technology to automate the attendance process.

Our system **detects and recognizes faces** using OpenCV and Face Recognition libraries, marking attendance automatically in an Excel sheet. This ensures accuracy, prevents proxy attendance, and saves time.

---

## 2. Objectives

The main goals of this project are:

- ✓ **Automate attendance marking** using face recognition.
  - ✓ **Ensure accuracy and prevent proxy attendance.**
  - ✓ **Speed up the process** compared to traditional methods.
  - ✓ **Store attendance records securely** in an Excel file.
  - ✓ **Prevent duplicate entries** by ensuring a person is marked once per day.
- 

## 3. Technologies Used

This project is implemented in **Python** using the following libraries:

Library	Purpose
opencv-python	Capturing and processing images from the webcam.
numpy	Handling numerical computations.
face_recognition	Face detection and encoding.
csv	Managing attendance records.
datetime	Fetching current date and time.
openpyxl	Converting CSV to Excel and adjusting column width.

To install the required libraries, run:

```
pip install opencv-python numpy face-recognition openpyxl
```

---

## 4. Working Mechanism

1❑. **Face Encoding:** The system first reads known faces from a folder (`faces/`) and extracts unique facial encodings.

2❑. **Live Face Detection:** The webcam captures a frame, detects faces, and extracts encodings.

3. **Face Matching:** The system compares detected faces with stored encodings to identify the person.

4❑. **Attendance Marking:** If a match is found and attendance hasn't been recorded for the day, the system logs the name, date, and time in `attendance.xlsx`.

5❑. **CSV to Excel Conversion:** The system automatically converts the attendance file from CSV to Excel and adjusts column widths to prevent formatting issues.

---

## 5. Code Explanation

### Step 1: Load Known Faces

```
path = 'faces'
images = []
classNames = []
face_list = os.listdir(path)

for cl in face_list:
    cur_img = cv2.imread(f'{path}/{cl}')
    images.append(cur_img)
    classNames.append(os.path.splitext(cl)[0])
```

✦ Reads images from the `faces/` folder and extracts names.

### Step 2: Encode Faces

```
def find_encodings(images):
    encode_list = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)
        if encode:
            encode_list.append(encode[0])
    return encode_list

known_encodings = find_encodings(images)
```

✦ Converts images to encodings for faster face matching.

### Step 3: Mark Attendance

```
def mark_attendance(name):
    now = datetime.now()
    date_today = now.strftime("%Y-%m-%d")
    time_now = now.strftime("%H:%M:%S")

    if name in marked_today:
        return

    with open("attendance.csv", 'a', newline='') as f:
        writer = csv.writer(f)
        writer.writerow([name, date_today, time_now])
    print(f"✔ {name}'s attendance marked!")
    marked_today.add(name)
```

✦ Ensures that attendance is marked only **once per day per person**.

## Step 4: Convert CSV to Excel

```
def csv_to_excel_and_delete(csv_file, excel_file):
    wb = openpyxl.Workbook()
    ws = wb.active
    ws.title = "Attendance"

    with open(csv_file, 'r') as f:
        reader = csv.reader(f)
        for row in reader:
            ws.append(row)

    for col in ws.columns:
        max_length = 0
        col_letter = col[0].column_letter
        for cell in col:
            if cell.value:
                max_length = max(max_length, len(str(cell.value)))
        ws.column_dimensions[col_letter].width = max_length + 2

    wb.save(excel_file)
    os.remove(csv_file)
```

✦ Fixes column width issue (#### in Excel) and **deletes the CSV after conversion**.

## Step 5: Face Detection & Attendance Marking

```
cap = cv2.VideoCapture(0)

while True:
    success, img = cap.read()
    img_small = cv2.resize(img, (0, 0), fx=0.25, fy=0.25)
    img_small = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB)

    faces_current_frame = face_recognition.face_locations(img_small)
    encodes_current_frame = face_recognition.face_encodings(img_small,
    faces_current_frame)

    for encodeFace, faceLoc in zip(encodes_current_frame,
    faces_current_frame):
        matches = face_recognition.compare_faces(known_encodings,
        encodeFace)
```

```
        faceDis = face_recognition.face_distance(known_encodings,
        encodeFace)

        if len(faceDis) > 0:
            matchIndex = np.argmin(faceDis)

            if matches[matchIndex]:
                name = classNames[matchIndex].upper()
                mark_attendance(name)

        cv2.imshow('Face Attendance System', img)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

✦ Captures real-time video, detects faces, and marks attendance **only once per day**.

---

## 6. Features of the System

- ✓ **Automatic Face Detection** – Recognizes faces in real time.
  - ✓ **Duplicate Prevention** – Ensures attendance is marked only once per day.
  - ✓ **Optimized for Speed** – Uses precomputed encodings for faster recognition.
  - ✓ **CSV to Excel Conversion** – Prevents formatting issues like ####.
  - ✓ **Simple and User-Friendly** – No manual input required.
- 

## 7. Future Enhancements

- ◆ **GUI Interface** (Tkinter/PyQt for better usability).
  - ◆ **Cloud Integration** (Store attendance records online).
  - ◆ **Mobile Support** (Use phone cameras for attendance).
- 

## 8. Conclusion

This **Face Recognition Attendance System** provides a **fast, accurate, and efficient** way to track attendance. By eliminating manual errors and preventing proxy attendance, this system is ideal for schools, offices, and institutions. With future enhancements like cloud storage and mobile integration, it can be further improved for large-scale deployment.

---